

Εργασία 2 – Ομαδική επικοινωνία με καθολική σειρά παράδοσης μηνυμάτων

Αναπτύξτε λογισμικό ομαδικής επικοινωνίας που παρέχει λειτουργικότητα αξιόπιστης καθολικής πολυεκπομπής (reliable total group multicast). Το λογισμικό θα πρέπει να προσφέρει ένα κατάλληλο API, στο πνεύμα των παρακάτω λειτουργιών:

<code>int grp_join(char *name, char *addr, int port, char *myid);</code>	εγγραφή σε ομάδα
<code>int grp_leave(int gsock);</code>	διαγραφή από ομάδα
<code>int grp_send(int gsock, void *msg, int len);</code>	αποστολή μηνύματος
<code>int grp_recv(int gsock, int *type, void *msg, int *len, int block);</code>	παραλαβή μηνύματος

Δεν επιτρέπεται να γίνουν υποθέσεις για το περιεχόμενο των μηνυμάτων της εφαρμογής. Επίσης, πρέπει να υπάρχει απεμπλοκή των εσωτερικών μηχανισμών του ενδιάμεσου λογισμικού από την εφαρμογή που δεν πρέπει να μπλοκάρει χωρίς λόγο. Για παράδειγμα, η αποστολή μηνυμάτων προς την ομάδα πρέπει να γίνεται ασύγχρονα. Εκτός από το ενδιάμεσο λογισμικό σε επίπεδο μέλους ομάδας πρέπει να υλοποιήσετε μια βοηθητική υπηρεσία διαχείρισης (βλέπε παρακάτω). Μπορείτε να υποθέσετε ότι οι διεργασίες/κόμβοι δεν παρουσιάζουν βλάβες.

Στάδιο 1: group management

Όταν μια διεργασία επιθυμεί να προσχωρήσει σε μια ομάδα, καλεί την `join()`. Εσωτερικά, στέλνονται στην υπηρεσία διαχείρισης τα στοιχεία της διεργασίας, και παραλαμβάνονται τα στοιχεία των μελών της ομάδας. Η υπηρεσία ειδοποιεί τα μέλη της ομάδας για την προσθήκη προτού στείλει επιβεβαίωση στο νέο μέλος. Τελικά η `grp_join()` επιστρέφει έναν ακέραιο στο πνεύμα ενός socket file descriptor, μέσω του οποίου η εφαρμογή μπορεί στη συνέχεια να στείλει/λάβει μηνύματα στην/από την ομάδα. Αντίστοιχα, η `grp_leave()` διαγράφει την διεργασία από την ομάδα στην υπηρεσία διαχείρισης, που ενημερώνει τα υπόλοιπα μέλη για την διαγραφή προτού επιβεβαιώσει την διαγραφή στο απερχόμενο μέλος. Ο εντοπισμός της υπηρεσίας διαχείρισης πρέπει να γίνεται με UDP multicast, ενώ η επικοινωνία με την υπηρεσία διαχείρισης πρέπει να γίνεται με TCP/IP. Οι αλλαγές σύνθεσης της ομάδας ανακοινώνονται στην εφαρμογή μέσω μηνυμάτων που δημιουργεί το ενδιάμεσο λογισμικό. Ο διαχωρισμός τους από τα μηνύματα της εφαρμογής γίνεται μέσω πληροφορίας τύπου, με βάση κατάλληλη σύμβαση που ορίζει και υλοποιεί το ενδιάμεσο λογισμικό. Επίσης, το ενδιάμεσο λογισμικό ορίζει και το περιεχόμενο των μηνυμάτων αλλαγής σύνθεσης έτσι ώστε αυτά να μπορεί να αποκωδικοποιούνται/ερμηνεύονται σωστά από την εφαρμογή.

Στάδιο 2: total multicast

Η εφαρμογή στέλνει / παραλαμβάνει μηνύματα μέσω `grp_send()` / `grp_recv()`. Το ενδιάμεσο λογισμικό πρέπει να εγγυάται ότι τα μηνύματα θα παραδοθούν με την ίδια σειρά σε όλα τα υφιστάμενα μέλη της ομάδας. Η μετάδοση μηνυμάτων πρέπει να γίνεται με UDP/IP, με εκμετάλλευση της δυνατότητας φυσικής πολυεκπομπής (UDP multicast). Υποθέστε ότι τα μηνύματα χωράνε σε ένα UDP datagram. Υποθέστε ότι η σύνθεση της ομάδας παραμένει σταθερή κατά την διάρκεια της διάδοσης μηνυμάτων. Μετρήστε την απόδοση της υλοποίησης μέσω μιας απλής εφαρμογής που στέλνει $K > 1$ συνεχόμενες φορές ένα μήνυμα που περιέχει ως πληροφορία έναν μειούμενο σειριακό αριθμό, και αφού στείλει το τελευταίο μήνυμα (με τιμή 0) παραλαμβάνει ένα μήνυμα επιβεβαίωσης από κάθε μέλος της ομάδας με περιεχόμενο τον αριθμό των μηνυμάτων που παρέλαβε. Καταγράψτε τον μέσο αριθμό μηνυμάτων που στέλνονται σε επίπεδο UDP/IP και την μέση καθυστέρηση μετάδοσης ενός μηνύματος, ως συνάρτηση του αριθμού N των μελών της ομάδας ($N=2,3,4,5$), αν όλες οι διεργασίες είναι στο ίδιο μηχάνημα και αν κάθε διεργασία είναι σε ένα ξεχωριστό μηχάνημα.

Στάδιο 3: view synchronous message delivery (προαιρετικά)

Επεκτείνετε την υλοποίηση σας έτσι ώστε να υποστηρίζει αλλαγές της ομάδας κατά την διάρκεια της μετάδοσης μηνυμάτων, στο πνεύμα του view-synchronous multicast. Σκεφτείτε αν/πως πρέπει να αλλάξει η αλληλεπίδραση μεταξύ των μελών της ομάδας καθώς και με την υπηρεσία διαχείρισης σύνθεσης της ομάδας, για να επιτευχθεί ο επιθυμητός συγχρονισμός.

Εφαρμογή τελικής επίδειξης

Δοκιμάστε/επιδείξτε το λογισμικό σας μέσω μιας εφαρμογής chat. Σκεφτείτε πως μπορείτε να προσομοιώσετε την απώλεια μηνυμάτων ή/και καθυστερήσεις στην μετάδοση/επεξεργασία των μηνυμάτων πάνω από ένα αξιόπιστο/γρήγορο δίκτυο.

Χρησιμοποιήστε όποια γλώσσα προγραμματισμού επιθυμείτε. Ο κώδικας σας πρέπει να μεταφράζεται και να εκτελείται κανονικά στο περιβάλλον Linux του εργαστηρίου. Εναλλακτικά, μπορείτε να επιδείξετε την υλοποίηση σε δικά σας laptop που θα πρέπει να συνδέονται στο ενσύρματο δίκτυο του εργαστηρίου. Σιγουρευτείτε ότι η επίδειξη σας δουλεύει σωστά.

Φροντιστήριο/συζήτηση: **Πέμπτη 14 Μαρτίου 2019**

Ημερομηνία παράδοσης: **Σάββατο 30 Μαρτίου 2019, 22:00**